

Charles Babbage's Register Transfer Level (RTL) Language

1826

Michael D. Godfrey

21 November 2021

email:michaeldgodfrey@gmail.com

1. Introduction

It appears not to be well-known that sometime before 1826 Babbage invented, implemented and made use of the first Register Transfer Level (RTL) Language. He published the definition of the language in 1826[1]. He specifically used this language to verify the correctness of his analytical engine. He discusses this in his letter to Lord Derby which begins on page 100 of his autobiography[2]. He makes clear that the analytical engine could not have been implemented without this formal verification language.

However, it has been widely thought among those who take a serious interest in digital design verification that Seymour Cray was the first inventor of an RTL language.

Recently (2011) a new republication of Babbage's autobiography has become available from Cambridge University Press. This digitally enhanced version is very much more readable than any others [2]. This is and the Babbage 1926 paper are the main sources for this note.

2. The Verification Language (RTL)

This provided the first formal language suitable for the definition and verification of digital computer logic at the register transfer level. However, it is certain that Cray developed his language using his own notation and without knowledge of the original definition of such a language by Charles Babbage in 1826.

Babbage's work is described in detail in his paper in the Royal Society Phil. Trans. *On a method of expressing by signs the action of machinery*[1]. Babbage used his defined RTL language to carry out the formal verification of the correct operation of the Analytical Engine as he completed its design and construction.

A somewhat more readable copy of the Royal Society paper is available at the Internet Archive:

<https://archive.org/details/babbageroyalsocvol116xviii/pg2502651826>.

Or click on: [Babbage](#).

The first four paragraphs of Babbage's paper read as follows:

2 Charles Babbage's Register Transfer Level (RTL) Language

“In the construction of an engine, on which I have now been for some time occupied, for the purpose of calculating tables and impressing the results on plates of copper, I experienced great delay and inconvenience from the difficulty of ascertaining from the drawings the state of motion or rest of any individual part at any given instant of time : and if it became necessary to enquire into the state of several parts at the same moment the labour was much encreased.

In the description of machinery by means of drawings, it is only possible to represent an engine in one particular state of its action. If indeed it is very simple in its operations, a succession of drawings may be made of it in each state of its progress which will represent its whole course ; but this rarely happens, and is attended with the inconvenience and expense of numerous drawings. The difficulty of retaining in the mind all the contemporaneous and successive movements of a complicated machine, and the still greater difficulty of properly timing movements which had already been provided for, induced me to seek for some method by which I might at a glance of the eye select any particular part, and find at any given time its state of motion or rest, its relation to the motions of any other part of the machine, and if necessary trace back the sources of its movement through all its successive stages to the original moving power. I soon felt that the forms of ordinary language were far too diffuse to admit of any expectation of removing the difficulty, and being convinced from experience of the vast power which analysis derives from the great condensation of meaning in the language it employs, I was not long in deciding that the most favourable path to pursue was to have recourse to the language of signs. It then became necessary to contrive a notation which ought if possible to be at once simple and expressive, easily understood at the commencement, and capable of being readily retained in the memory from the proper adaptation of the signs to the circumstances they were intended to represent. The first thing to be done was obviously to make an accurate enumeration of all the moving parts, and to appropriate a name to each ; the multitude of different contrivances in various machinery, precluded all idea of substituting signs for these parts. They were therefore written down in succession, only observing to preserve such an order that those which jointly concur for accomplishing the effect of any separate part of the machine might be found situated near to each other : thus in a clock, those parts which belong to the striking part ought to be placed together, whilst those by which the repeating part operates ought, although kept distinct, yet to be as a whole, adjacent to the former part.

Each of these names is attached to a faint line which runs longitudinally down the page, and which may for the sake of reference be called its *indicating line*.

The next object was to connect the notation with the drawings of the machine, in order that the two might mutually illustrate and explain each other.”

3. RTL Language Development

Seymour Cray is often referenced as the inventor of the first RTL language. He was a key designer of the ERA 1103 in the early 1950's.

The details of his design contribution have not been recorded. In discussions with the UNIVAC engineers in Roseville who had been with the ERA company during the 1950's I learned that one day Cray appeared with pages of written notes. He described these as the formal definition at the register level of the ERA 1103. This provided a formal notation for the definition of digital logic at the register transfer level. It is certain that Cray invented this language using his own notation without knowledge of Babbage's original definition of such a language.

The Cray RTL was used to verify the 1103.

This system became the UNIVAC 1103 announced in February 1953. (Click on: [UNIVAC 1103](#).) Or see: (https://en.wikipedia.org/wiki/UNIVAC_1103#cite_note-2).

A compiler for the language was subsequently implemented and this became a key element in the design verification for subsequent UNIVAC and UNISYS computer systems. This continued until a unified CMOS VLSI design system was put in place. This incorporated the verification logic from the previous system.

Of course by now such languages are widely used by digital circuit designers.

References

- [1] Babbage, C.: On a method of expressing by signs the action of machinery. *Royal Society* 116-XVIII, 250–265 (1826). Available at: royalsocietypublishing.org or at the [Internet Archive](#).
- [2] Babbage, C.: *Passages from the Life of a Philosopher*. Cambridge University Press, New York (1864). Be sure to obtain this **digitally printed version 2011** from: [Cambridge Press](#)